Appointment System DB Management Assignment

Database Schema

Tables:

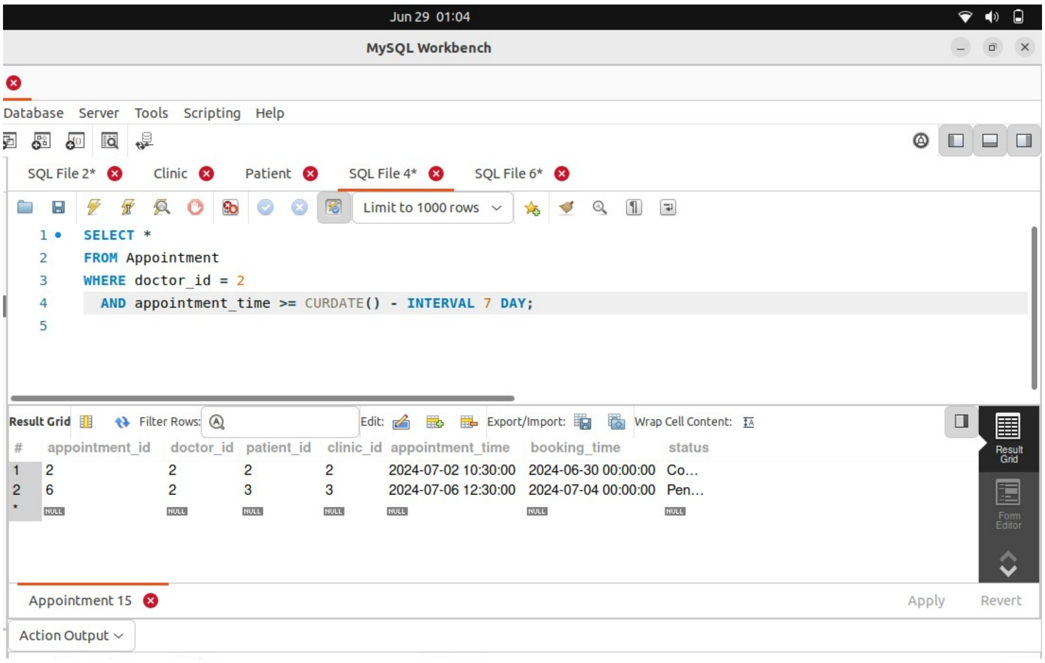
1. **Doctor**
   - `doctor_id` INT PRIMARY KEY
   - `doctor_name` VARCHAR(100)
   - Other relevant details (e.g., specialty)
2. **Clinic**
   - `clinic_id` INT PRIMARY KEY
   - `clinic_name` VARCHAR(100)
   - Other relevant details (e.g., location)
3. **Patient**
   - `patient_id` INT PRIMARY KEY
   - `patient_name` VARCHAR(100)
   - `date_of_birth` DATE
   - Other relevant details (e.g., contact information)
4. **Appointment**
   - `appointment_id` INT PRIMARY KEY
   - `doctor_id` INT FOREIGN KEY REFERENCES Doctor(doctor_id)
   - `patient_id` INT FOREIGN KEY REFERENCES Patient(patient_id)
   - `clinic_id` INT FOREIGN KEY REFERENCES Clinic(clinic_id)
   - `appointment_time` DATETIME
   - `booking_time` DATE
   - `status` VARCHAR(20) (e.g., 'Confirmed', 'Pending', 'Cancelled')

ER Diagram:-

```
+-----------------+     +-------------------+
  |    Doctor     |     |     Clinic      |
+-----------------+     +-------------------+
| doctor_id (PK)  |     | clinic_id (PK)    |
| doctor_name     |     | clinic_name       |
      | ...       |     | ...             |
+-----------------+     +-------------------+
               |                  |
               |                  |
               |                  |
          +-------------------------+
                        |
                        |
                        |
              +---------------------+
              |    Appointment      |
              +---------------------+
            | appointment_id (PK)  |
              | doctor_id (FK)      |
              | patient_id (FK)     |
              | clinic_id (FK)      |
             | appointment_time     |
              | booking_time        |
                | status            |
```

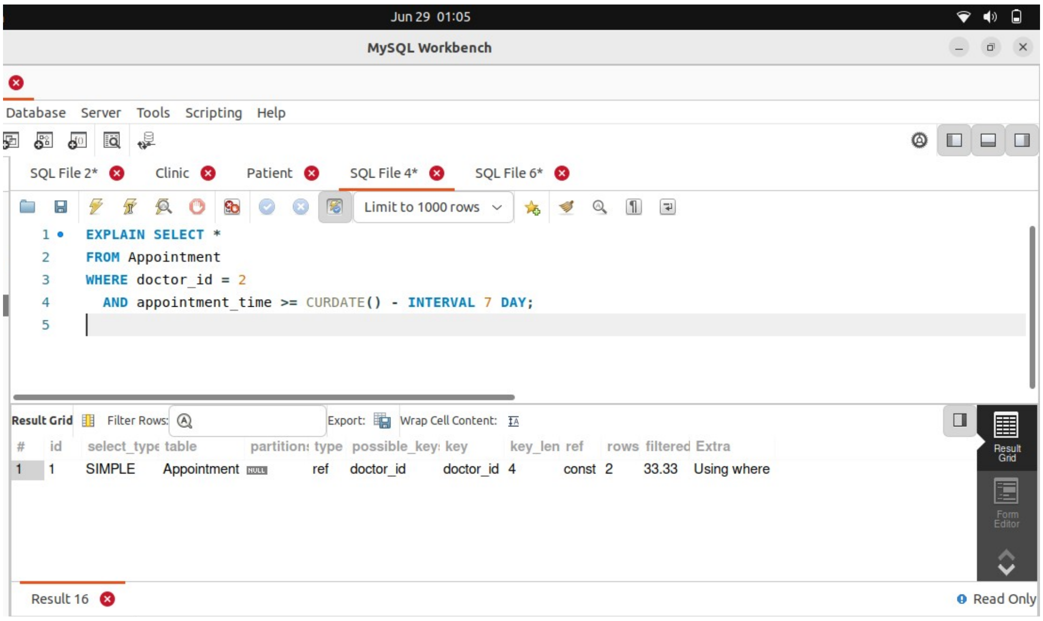
+---------------------+

QUERY 1:


Jun 29 01:04
MySQL Workbench
Database Server Tools Scripting Help
SQL File 2* Clinic Patient SQL File 4* SQL File 6*
Limit to 1000 rows
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
Result Grid
Form Editor
Appointment 15
Apply Revert
Action Output

```
SELECT *
FROM Appointment
WHERE doctor_id = 2
  AND appointment_time >= CURDATE() - INTERVAL 7 DAY;
```

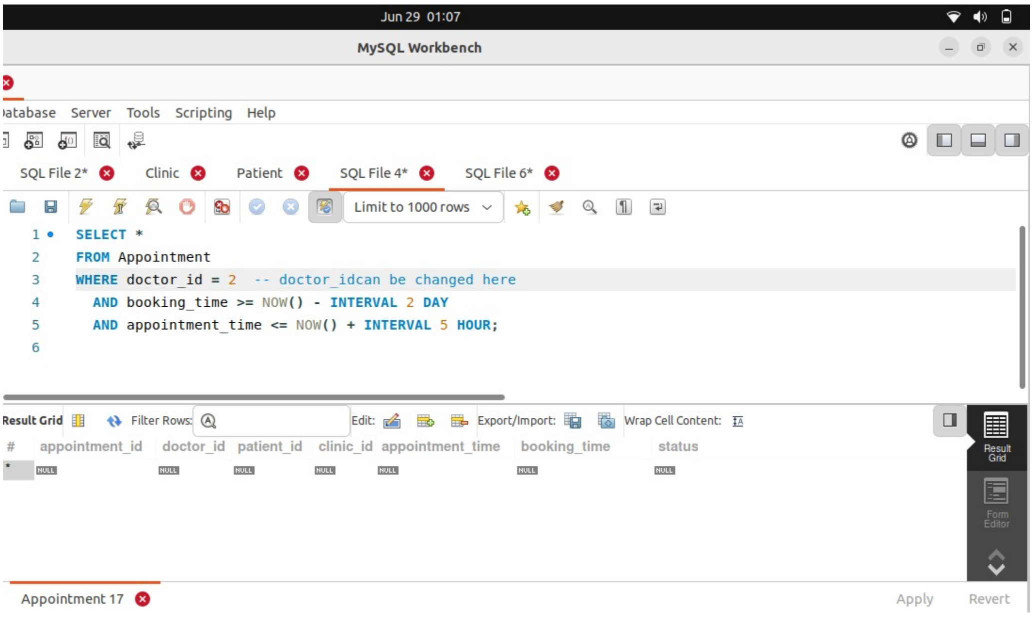
#	appointment_id	doctor_id	patient_id	clinic_id	appointment_time	booking_time	status
1	2	2	2	2	2024-07-02 10:30:00	2024-06-30 00:00:00	Co...
2	6	2	3	3	2024-07-06 12:30:00	2024-07-04 00:00:00	Pen...
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL


Jun 29 01:05
MySQL Workbench
Database Server Tools Scripting Help
SQL File 2* Clinic Patient SQL File 4* SQL File 6*
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content:
Result Grid
Form Editor
Result 16
Read Only

```
EXPLAIN SELECT *
FROM Appointment
WHERE doctor_id = 2
  AND appointment_time >= CURDATE() - INTERVAL 7 DAY;
```

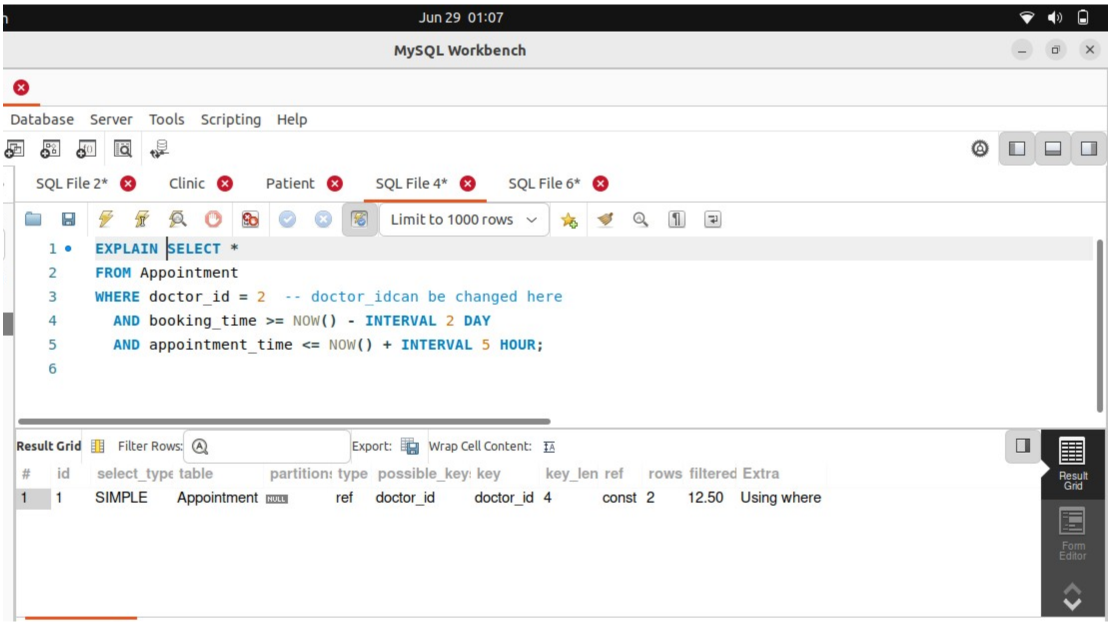
#	id	select_type	table	partition:	type	possible_key:	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	Appointment	NULL	ref	doctor_id	doctor_id	4	const	2	33.33	Using where

QUERY 2:


Jun 29 01:07
MySQL Workbench
Database Server Tools Scripting Help
SQL File 2* Clinic Patient SQL File 4* SQL File 6*
Limit to 1000 rows
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
Result Grid
Form Editor
Appointment 17
Apply Revert

```
SELECT *
FROM Appointment
WHERE doctor_id = 2  -- doctor_idcan be changed here
  AND booking_time >= NOW() - INTERVAL 2 DAY
  AND appointment_time <= NOW() + INTERVAL 5 HOUR;
```

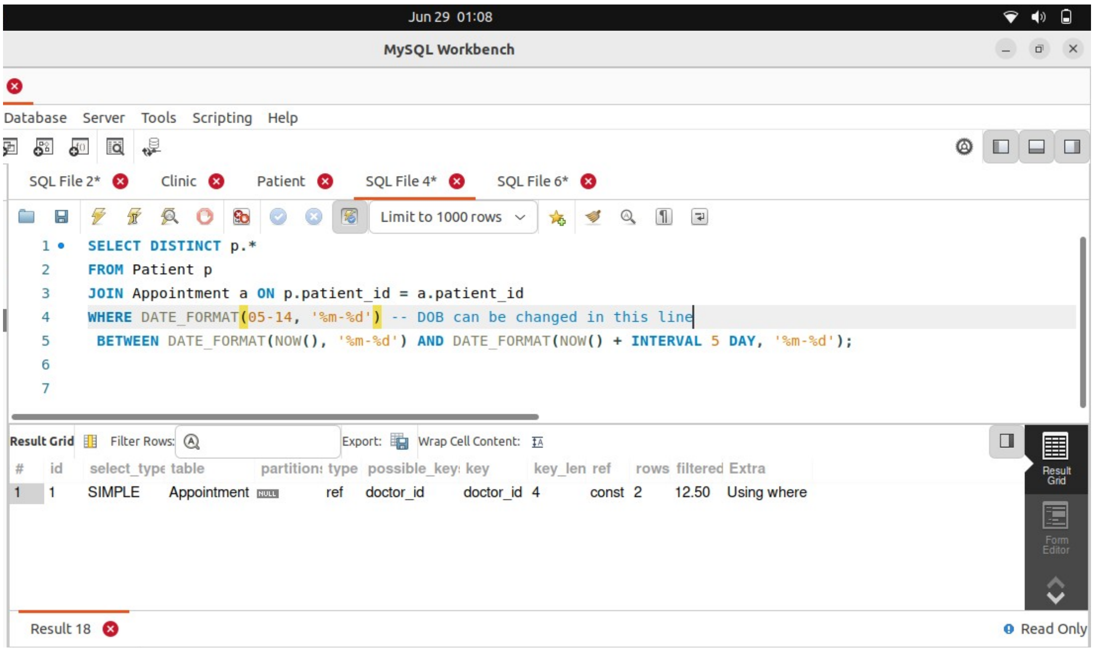
#	appointment_id	doctor_id	patient_id	clinic_id	appointment_time	booking_time	status
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL


Jun 29 01:07
MySQL Workbench
Database Server Tools Scripting Help
SQL File 2* Clinic Patient SQL File 4* SQL File 6*
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content:
Result Grid
Form Editor

```sql
EXPLAIN SELECT *
FROM Appointment
WHERE doctor_id = 2  -- doctor_idcan be changed here
  AND booking_time >= NOW() - INTERVAL 2 DAY
  AND appointment_time <= NOW() + INTERVAL 5 HOUR;
```

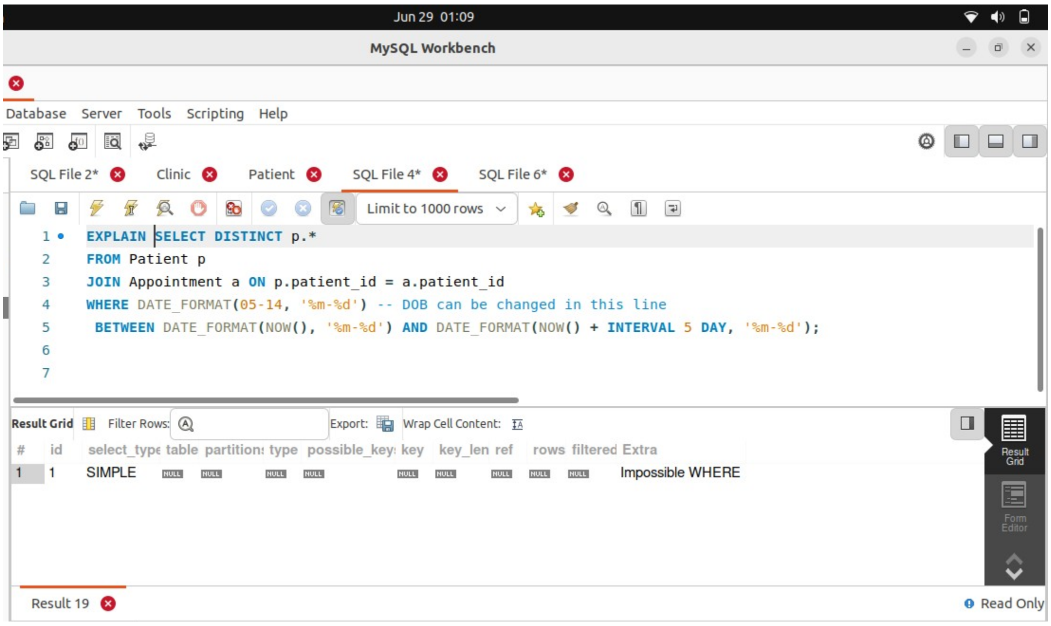
#	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	Appointment	NULL	ref	doctor_id	doctor_id	4	const	2	12.50	Using where

QUERY 3:


Jun 29 01:08
MySQL Workbench
Database Server Tools Scripting Help
SQL File 2* Clinic Patient SQL File 4* SQL File 6*
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content:
Result Grid
Form Editor
Result 18
Read Only

```sql
SELECT DISTINCT p.*
FROM Patient p
JOIN Appointment a ON p.patient_id = a.patient_id
WHERE DATE_FORMAT(05-14, '%m-%d') -- DOB can be changed in this line
 BETWEEN DATE_FORMAT(NOW(), '%m-%d') AND DATE_FORMAT(NOW() + INTERVAL 5 DAY, '%m-%d');
```

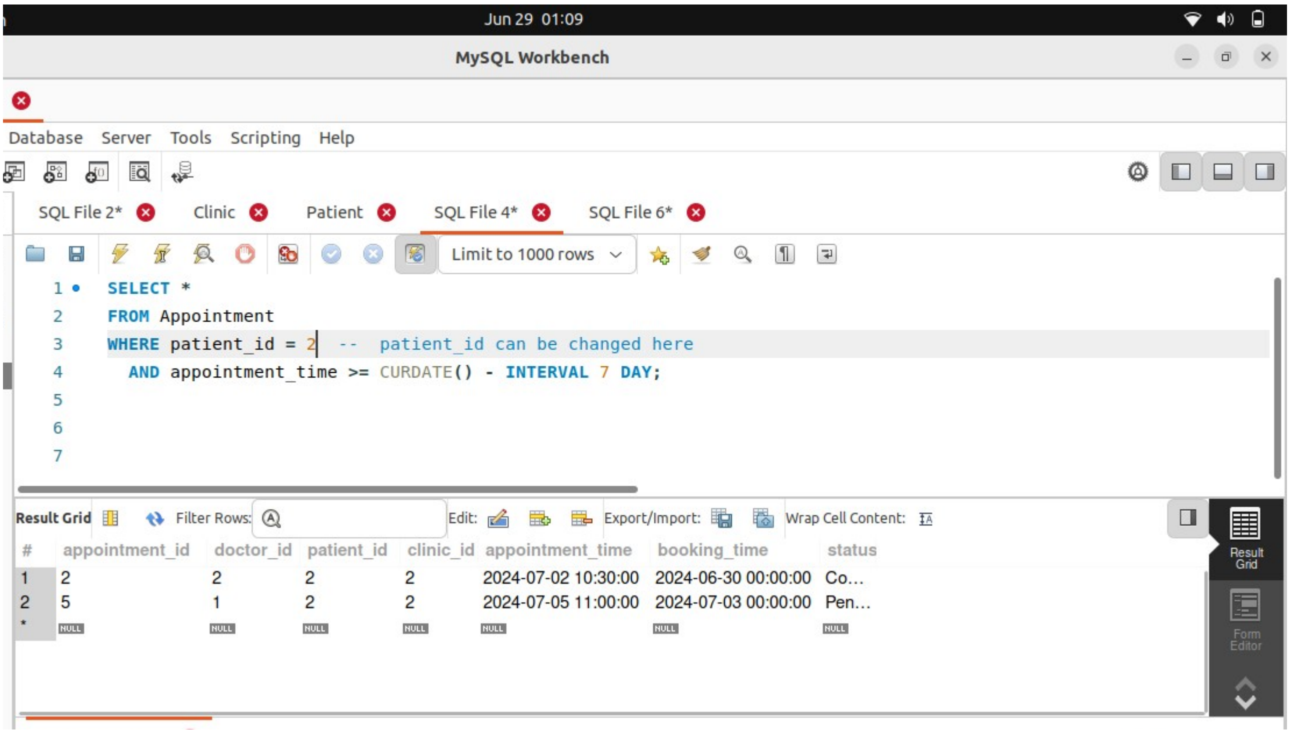
#	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	Appointment	NULL	ref	doctor_id	doctor_id	4	const	2	12.50	Using where


Jun 29 01:09
MySQL Workbench
Database Server Tools Scripting Help
SQL File 2* Clinic Patient SQL File 4* SQL File 6*
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content:
Result Grid
Form Editor
Result 19
Read Only

```sql
EXPLAIN SELECT DISTINCT p.*
FROM Patient p
JOIN Appointment a ON p.patient_id = a.patient_id
WHERE DATE_FORMAT(05-14, '%m-%d') -- DOB can be changed in this line
 BETWEEN DATE_FORMAT(NOW(), '%m-%d') AND DATE_FORMAT(NOW() + INTERVAL 5 DAY, '%m-%d');
```

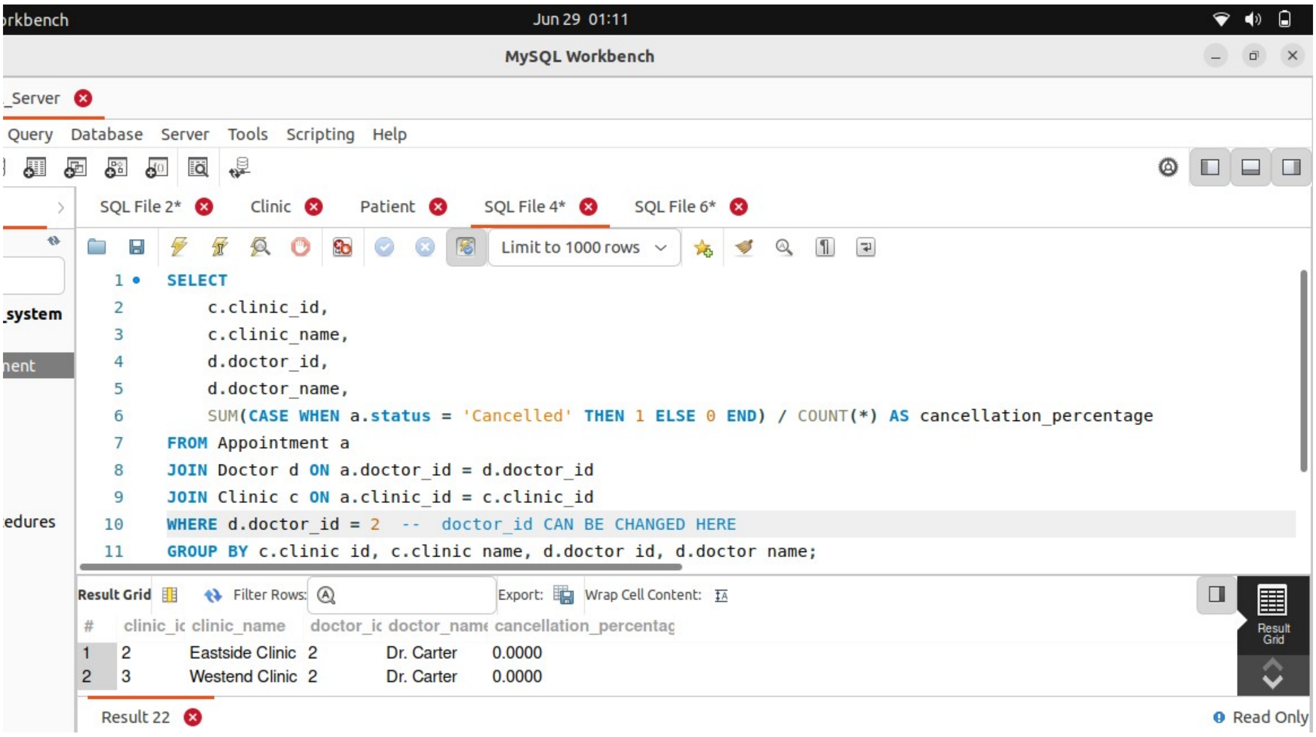
#	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	Impossible WHERE

QUERY 4:


Jun 29 01:09
MySQL Workbench
Database Server Tools Scripting Help
SQL File 2* Clinic Patient SQL File 4* SQL File 6*
Limit to 1000 rows
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
Result Grid
Form Editor

```sql
SELECT *
FROM Appointment
WHERE patient_id = 2  -- patient_id can be changed here
  AND appointment_time >= CURDATE() - INTERVAL 7 DAY;
```

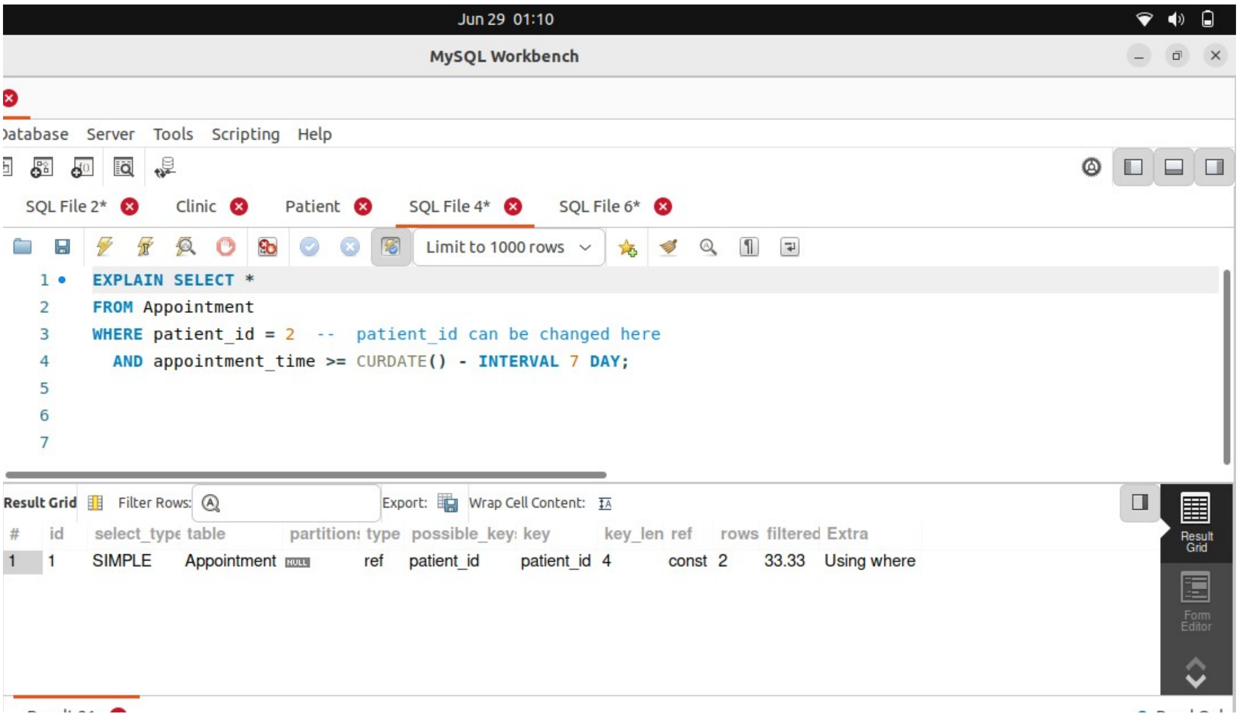
#	appointment_id	doctor_id	patient_id	clinic_id	appointment_time	booking_time	status
1	2	2	2	2	2024-07-02 10:30:00	2024-06-30 00:00:00	Co...
2	5	1	2	2	2024-07-05 11:00:00	2024-07-03 00:00:00	Pen...
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL


rkbench
Jun 29 01:11
MySQL Workbench
_Server
Query Database Server Tools Scripting Help
SQL File 2* Clinic Patient SQL File 4* SQL File 6*
_system
nent
edures
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content:
Result Grid
Result 22
Read Only

```sql
SELECT
    c.clinic_id,
    c.clinic_name,
    d.doctor_id,
    d.doctor_name,
    SUM(CASE WHEN a.status = 'Cancelled' THEN 1 ELSE 0 END) / COUNT(*) AS cancellation_percentage
FROM Appointment a
JOIN Doctor d ON a.doctor_id = d.doctor_id
JOIN Clinic c ON a.clinic_id = c.clinic_id
WHERE d.doctor_id = 2  --  doctor_id CAN BE CHANGED HERE
GROUP BY c.clinic id, c.clinic name, d.doctor id, d.doctor name;
```

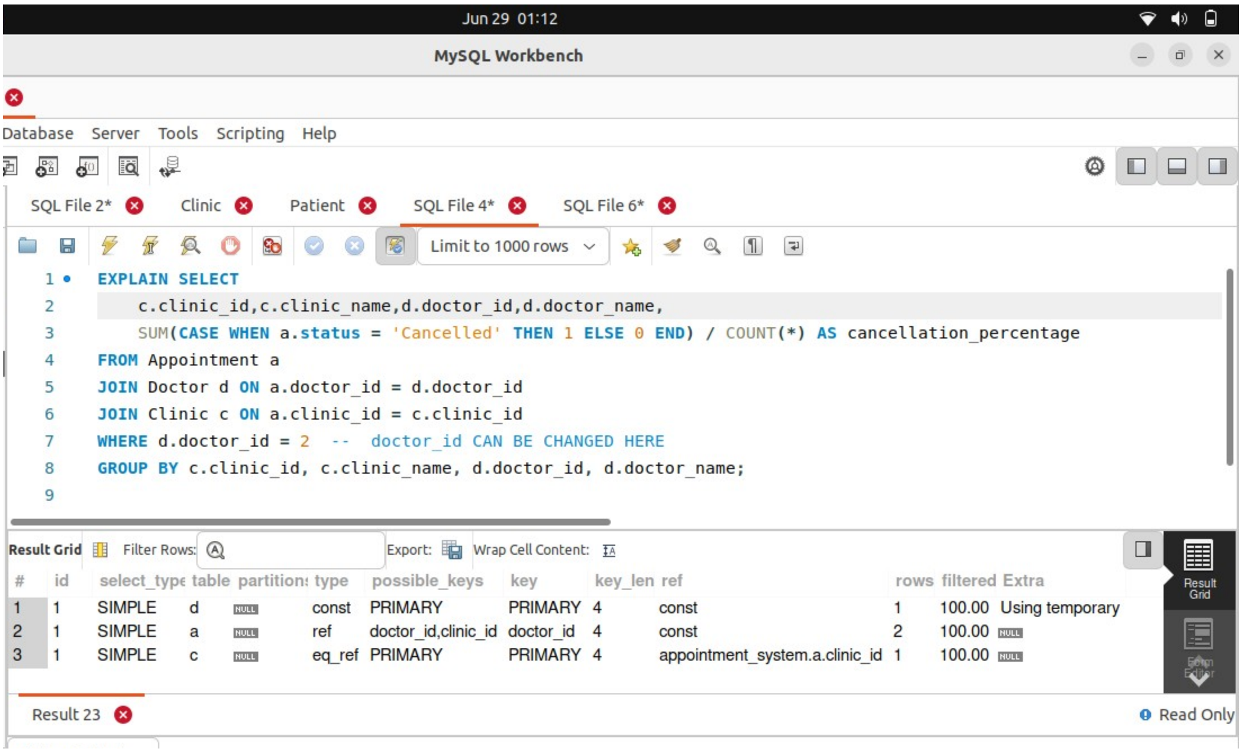
#	clinic_ic	clinic_name	doctor_ic	doctor_name	cancellation_percentag
1	2	Eastside Clinic	2	Dr. Carter	0.0000
2	3	Westend Clinic	2	Dr. Carter	0.0000

QUERY 5:


Jun 29 01:10
MySQL Workbench
Database Server Tools Scripting Help
SQL File 2* Clinic Patient SQL File 4* SQL File 6*
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content:
Result Grid
Form Editor

```sql
EXPLAIN SELECT *
FROM Appointment
WHERE patient_id = 2  --  patient_id can be changed here
  AND appointment_time >= CURDATE() - INTERVAL 7 DAY;
```

#	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	Appointment	NULL	ref	patient_id	patient_id	4	const	2	33.33	Using where


Jun 29 01:12
MySQL Workbench
Database Server Tools Scripting Help
SQL File 2* Clinic Patient SQL File 4* SQL File 6*
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content:
Result Grid
Form Editor
Result 23
Read Only

```sql
EXPLAIN SELECT
    c.clinic_id,c.clinic_name,d.doctor_id,d.doctor_name,
    SUM(CASE WHEN a.status = 'Cancelled' THEN 1 ELSE 0 END) / COUNT(*) AS cancellation_percentage
FROM Appointment a
JOIN Doctor d ON a.doctor_id = d.doctor_id
JOIN Clinic c ON a.clinic_id = c.clinic_id
WHERE d.doctor_id = 2  --  doctor_id CAN BE CHANGED HERE
GROUP BY c.clinic_id, c.clinic_name, d.doctor_id, d.doctor_name;
```

#	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	d	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	Using temporary
2	1	SIMPLE	a	NULL	ref	doctor_id,clinic_id	doctor_id	4	const	2	100.00	NULL
3	1	SIMPLE	c	NULL	eq_ref	PRIMARY	PRIMARY	4	appointment_system.a.clinic_id	1	100.00	NULL